Algebraic and Transcendental Equations

G V V Sharma*

Abstract—**Through examples, this manual introduces methods for solving algebraic and transcendental equations like the bisection method, the method of false position, the iteration method and the Newton-Raphson method Python codes are provided for all these methods.**

1 Fixed Point Iteration

Theorem 1.1. Consider the equation

$$x = g(x) \tag{0.1}$$

If $|g'(x) \leqslant K < 1|$, then it is possible to frame a difference equation

$$x_{n+1} = g(x_n) \tag{0.2}$$

to solve (0.1) [1].

Problem 1. Sketch

$$x^3 + x - 1 = 0 \tag{1.1}$$

Solution: The following code generates the Fig. 1

```
import numpy as np
import matplotlib.pyplot as plt



x = np.linspace(-5,5,20)
y = x**3+x-1
plt.plot(x,y)
plt.grid()
plt.xlabel('$x$')
plt.ylabel('$y$')
#plt.savefig('../figs/fpi_example.
    eps')
plt.show()
```

*The author is with the Department of Electrical Engineering, IIT, Hyderabad 502285 India e-mail: gadepall@iith.ac.in.

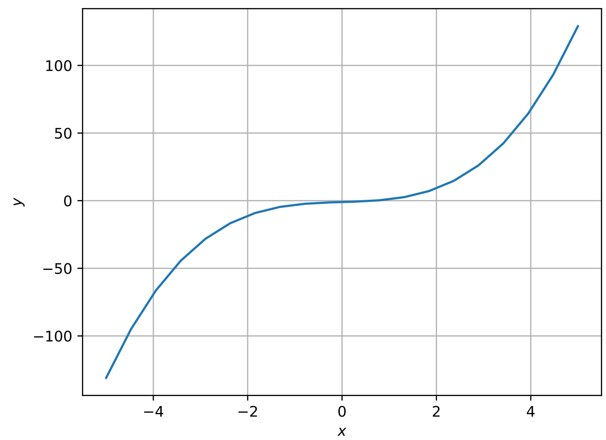

Fig. 1: The solution is at the point where the function meets the X-axis.

Problem 2. Obtain the difference equation for (1.1) by iteration.

Solution: Expressing (1.1) as

$$x = \frac{1}{1 + x^2}, \tag{2.1}$$

the difference equation

$$x_{n+1} = \frac{1}{1 + x_n^2}, \tag{2.2}$$

is obtained.

Problem 3. Solve (2.2).

Solution: The following script computes (2.2) resulting in

$$x = 0.6823278038681895 \tag{3.1}$$

```
import numpy as np
import matplotlib.pyplot as plt

def g(x):
```

```
        return  1/(1+x**2)

a = 1
n = 50
for i in range(n):
        a = g(a)

print(a)
```

2 Newton-Ralphson Method

Problem 4. Obtain the difference equation for solving

$$f(x) = 0. \tag{4.1}$$

Solution: From [2], the desired difference equation is

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)} \tag{4.2}$$

Problem 5. Solve (1.1) using (4.2) for 4 iterations.

Solution: Using the fact that

$$f'(x) = 3x^2 + 1, \tag{5.1}$$

The following script computes (4.2) resulting in

$$x = 0.6823278039465127 \tag{5.2}$$

```
import numpy as np
import matplotlib.pyplot as plt

def f(x):
        return x**3 +x -1

def f1(x):
        return 3*x**2 + 1

a = 1
n = 4
for i in range(n):
        a = a - f(a)/f1(a)

print(a)
```

Problem 6. Compare the Newton-Ralphson method with the Iteration method.

3 Bisection Method

Theorem 3.1. Consider (1.1) and Fig. 1. Choose a points a, b for which $f(a) < 0, f(b) > 0$. Find $c_0 = \frac{a+b}{2}$. If $f(c_0) < 0, c_1 = \frac{c_1+b}{2}$, else $c_1 = \frac{a+c_1}{2}$. Similarly choose $c_2, c_3, \ldots$. For sufficiently large iterations, c_n will coverge to a root of $f(x)$. Note that this will happen only if $f(x)$ is continuous in (a, b) [3].

Problem 7. Solve (1.1) using Theorem 3.1.

Solution: From Fig. 1, $f(-1) < 0$ and $f(1) > 0$. Choosing $a = -1, b = 1$ initially, the following script computes the solution resulting in

$$x = 0.6823278038280183 \tag{7.1}$$

for 50 iterations.

```
import numpy as np
import matplotlib.pyplot as plt

def f(x):
        return x**3 +x -1

a = -1
b = 1

n = 50

for i in range(n):
        c = (a + b)/2
        if f(c) < 0 :
                a = c
        else:
                b = c


print(c)
```

4 Method of False Position

Theorem 4.1. Consider (1.1). Choose a points a, b for which $f(a) < 0, f(b) > 0$. Find [4]

$$c_k = b_k - f(b_k)\frac{(b_k - a_k)}{f(b_k) - f(a_k)} \tag{7.2}$$

If $f(c_k) < 0, a_k = c_k$, else $b_k = c_k$. For sufficiently large iterations, c_n will coverge to a root of $f(x)$. Note that this will happen only if $f(x)$ is continuous in (a, b).

Problem 8. Solve (1.1) using Theorem 4.1.

Solution: From Fig. 1, $f(-1) < 0$ and $f(1) > 0$. Choosing $a = -1, b = 1$ initially, the following script computes the solution resulting in

$$x = 0.6823278038280193 \tag{8.1}$$

for 50 iterations.

```
import numpy as np
import matplotlib.pyplot as plt

def f(x):
        return x**3 +x -1

a = -1
b = 1

n = 50

for i in range(n):
        c = b - f(b)*(b-a)/(f(b)-f
            (a))
        if f(c) < 0 :
                a = c
        else:
                b = c


print(c)
```

Problem 9. What is the difference between the bisection method and the method of false position?

References

[1] E. Kreyszig, *Advanced engineering mathematics*, 8th ed. New Delhi,: Wiley,, c2007.
[2] Wikipedia. [Online]. Available: https://en.wikipedia.org/wiki/Newton%27s_method
[3] ——. [Online]. Available: https://en.wikipedia.org/wiki/Bisection_method
[4] ——. [Online]. Available: https://en.wikipedia.org/wiki/False_position_method